a **web of things** for smarter cities

As a PhD student at ETH Zurich and SAP, I have been exploring the Internet of Things (IoT), which is the vision of a world where all electronic devices are connected together to form a single, coherent network of heterogeneous devices. In theory, such a large network could collect real-time data from the physical world that can be used to solve all our problems, improve traffic in large cities, reduce pollution and energy consumption, take better care of the elderly and so on.

All this sounds nice in theory, but the reality is unfortunately different.

Early in my research, I realized that the Internet of Things is merely a scam - it is a leaky concept. On the one hand we have academic research that is not really concerned with standardization issues, therefore many incompatible solutions/projects have been explored by diverse research groups pretty much in isolation. On the other hand, we have the industry which is very interested by standards, as many companies that want to lead the IOT, so hundreds of protocols to communicate with devices have been created, proposed, even standardized. But are these standards really used? I mean by more than a few thousands IT experts in a very specific domain?
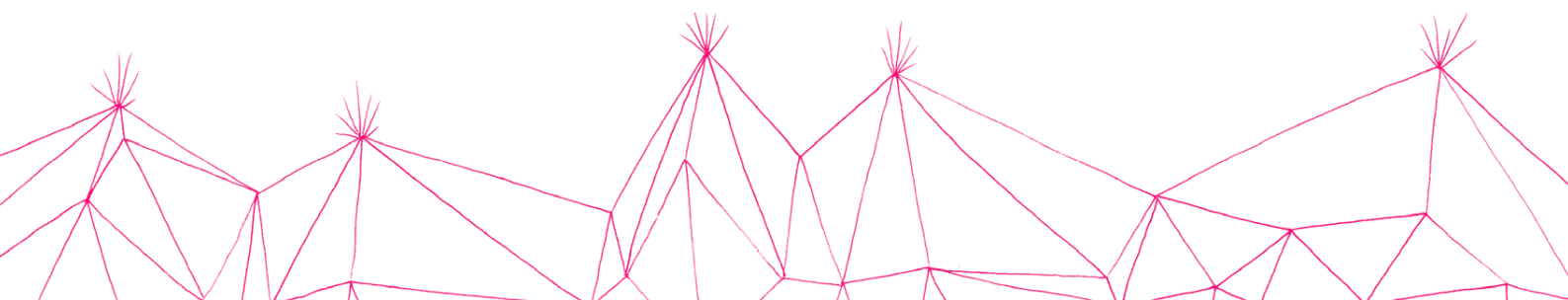
DPWS

Megaco   Modbus

OGC SensorML

Zorro III

AMQP

OPC-UA

UPnP

Let me illustrate this with a small sample of such standards commonly used in home automation and machine-to-machine communication. How many of you know and can develop using one of these protocols? Humm... yeah... This is exactly what I mean by leaky concept. Such a messy world where all of these protocols - or non-standard standards as I like to call them - co-habit, cannot become a unique Internet of Things. The reality today is that we have built many Intranets of things. Yes they work and do the job, but they remain isolated islands of a few connected devices. These islands have practically no way to interact with each other, and this observation reduces the classic vision of Internet of Things to merely an utopia. The status quo, makes it hard to share and reuse solutions, as for each new deployment tons of things need to be developed over again from scratch.

This is a huge waste of time and resources.

In fact, there is already a single standard out there to bind them all. We all know it and use it everyday: the Web. It is efficient and works well, and the reason the Web has become so successful is because it was free, open, flexible, and designed to be massively scalable. But above all, the main reason it has worked is because it was so simple, open, free, so anyone could use it. You became part of the Web by designing your first Web page using Frontpage and cheesy animated GIFs. Anyone could be part of it. All you needed was a computer, a modem and a text editor. This aspect is so fundamental that with my colleague Dom, we have explored how to adapt this magic recipe for electronic devices. They have a right to be part of the Web too, and as I will explain in this talk there are many reasons why they should be. Beyond obvious business opportunities by web-enabling devices, we have accumulated two decades of knowledge in building massively scalable, secure, and efficient Web sites. We have built a wealth of expertise for distributed caches, replicative DBs and so on. Why not reuse all that for devices too? Why do we need to reinvent wheels?
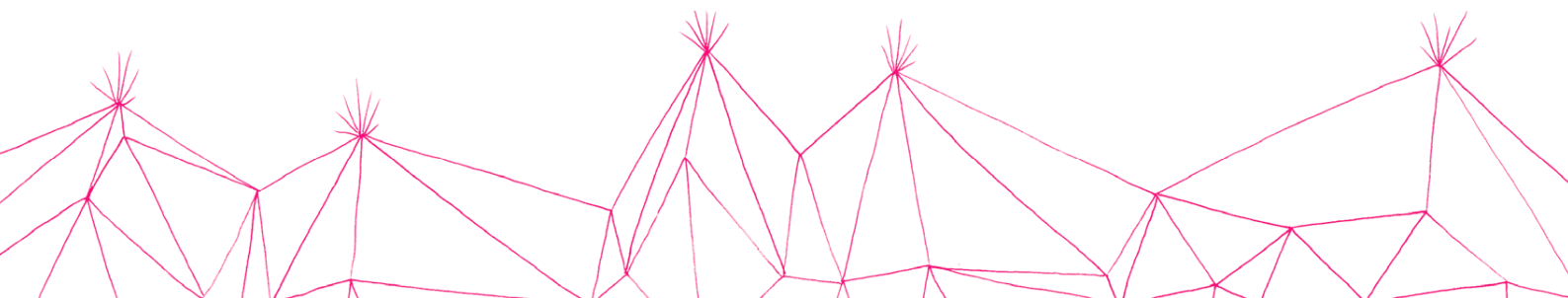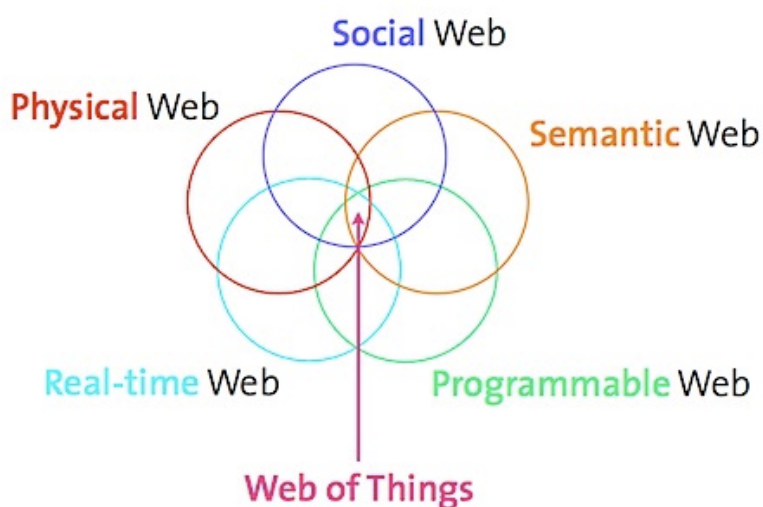
My argument here is that there is not a single world wide web, but 5 of them. We can see them as various trends, or facets of the Web as we know it today, and we all know and understand them. But, at the intersection of these 5W, when all these pieces are put together, a whole new, unknown territory emerges, and brand new possibilities to solve old problems are unlocked. This design space there is what we call the Web of Things. But first, let's first focus a little more on the physical Web and programmable Web.

Once upon a time, electronics and programming was reserved to the highest social class of our civilization, geeks. However, one day [processing](link) and [Arduino](link) appeared. This explosive cocktail has revealed to the world the conspiracy nerds were setting up, by demystifying our virtues and showed to the world how easy it actually is to program. Since then, countless designers and other people who were not meant to touch technology were suddenly empowered to create digital artifacts easily, on their own.These tools are an incredible example of how simple tools can democratize programming, by lowering the barrier for fast prototyping physico-digital artifacts.

How many of you can program processing/arduino?Not enough.

We want more people to be able to access and use real-time data. Not just raw sensor readings. I'm talking about data people care about. Data that can make our lives easier. Data that could make us happier. We believe the Web of Things to be the next evolution of the Web by enabling the democratization of programming and active citizenship. We believe this because the Web offers the lowest possible access barrier to simply create something that you can share with the world. Now how many of you have already made a Web page? Or setup a blog? Get my point? Everyone is a potential developer for the Web of Things. And you will be able to access and integrate real-time data from all kinds of sensors, simply by pasting some HTML code on your web page. I would like to share the vision of a large ecosystem of ubiquitous digital services roaming around, freely accessible. Imagine an ecosystem of reusable and shareable sensors, devices, and services that can be accessed simply via a Web API, using simply your facebook login information. Just by sending HTTP POST/GET request to a device, you read a sensor, you open a door. There are two ways to interact with this ecosystem: 1) "READ": collect data they record automatically and analyze it, or 2) "WRITE": pop up your phone, browse the space you're in for example query for an empty restaurant, call a cab, turn on lights, music or AC.

Mobile ad-hoc interaction (aka niwea)

Let's start directly the WRITE aspect first through an example. Imagine you go to a hotel in Japan, it's freezing and you're handed a remote like this. Chances are the accompanying translation such as this one is not included. What do you do? Why can't you just pop out your phone and control the AC directly with it? Or setup your alarm or book a tennis course offered by the hotel? Or even better, discretely ordering some champagne and caviar for your lady and play some romantic music while dimming the light? In a hotel room in Japan you've never been!!
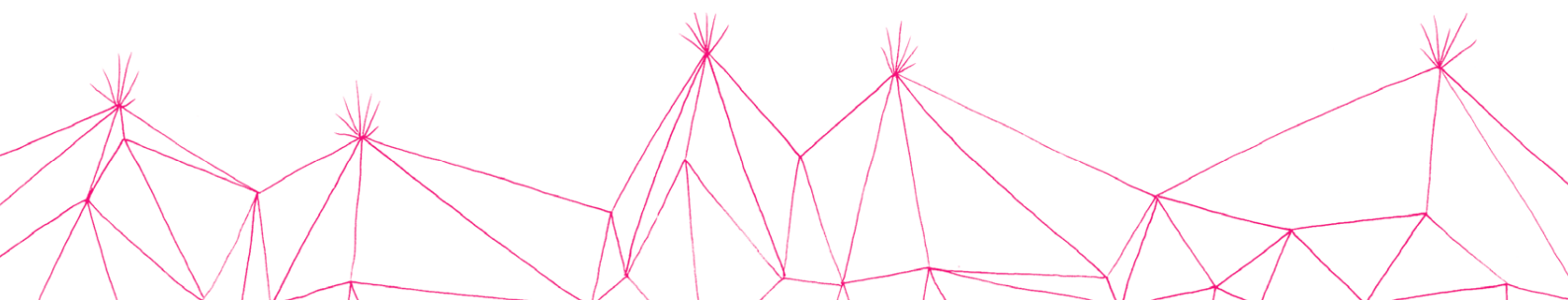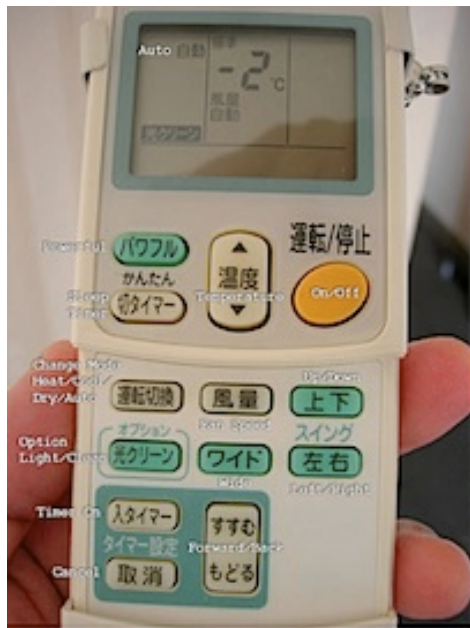
or this?

Now... what if I told you that for less than 10 dollars, you can buy a simple chip with a Wi-Fi antenna that could turn anything into a Web server? This means, any electronic thing can be connected to the Web and can be controlled via a Web API, and you would literally browse around the "physical" page of your room and find about the things in the physical world you could control? And just like most Websites in the world, it would recognize the language your phone and give you the room page in this same language.

Niwea is a term coined out by my friend hannes gassert, and stands for Native Interoperable WEb Apps (I wrote a bigger post about this earlier). If you're in the software industry, chances are a client or your boss told you "we need an iphone/symbian/windows mobile/android app". Of course, many self-respecting developers might reply "no, you don't really. You can do this apponly need a simple HTML/Javascript app that uses CSS style sheets to render your applications on these diverse mobile phones". So you only develop once a Web app, and it'll run on many devices. After this simple suggestion, the contra-argument generally goes along the lines of "bla bla want iphone app bla bla!!". Sencha, jQTouch, iui are some of those frameworks. They are not great yet, but it's only the beginning. Think about it: how many of you have made an iPhone app? Should I ask again about Web apps? iPhone developers: few and expensive (won't even mention code maintenance), Web developers: cheap & easy (and we got lots of 'em).

If you need performance or support for the native platform (GPS, camera, audio, etc), the N of niwea kicks in. Using phonegap, you can transform your Web app into a real, native app that will look, feel, and work just like the real thing. Only that it took you a few hours and $$ to make.

I do believe many more frameworks for fast prototyping Web apps (especially mobile) will appear and mature in the next years. Hopefully, more complete IDEs such as Flash) for developing interactive mobile Web apps with a few clicks. There is so much potential there!

Real-time cities

Now let's look at the other interaction mode, the "READ" aspect. O'Reilly has just organized STRATA, a conference that looks into data science - that is how to address this big question: how to efficiently analyze tons of raw data to extract meaningful information that could improve business process, marketing, etc. But what if the data in question is physical data? I spent the last couple of months exploring this possibility with the Senseable City Lab at MIT, on a project called LIVE Singapore! Let's take a city, for example Singapore. Lot of digital data traces are generated there every second via cameras, sensors of all kinds, radars & electronic road pricing, people with mobile phones, etc.

All this data represents a goldmine for everyone, if only it were to be used. Unfortunately, different companies collect it only to ensure everything works fine, and that's it. It's then stored behind closed doors - or worse, deleted - but rarely thoroughly analyzed.
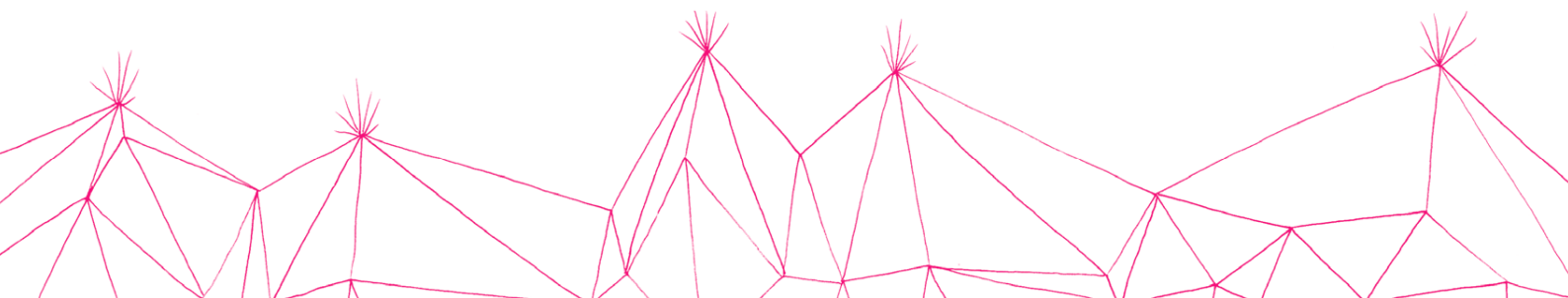
If only one could access it, so much valuable information could be extracted from it – valuable both for the company who could improve their processes or optimize their operations and citizens at large.
Optimizing water used to irrigate parks, managing the lighting in a smart way, providing an information system of free parking spaces or water leaks in pipes are problems common to most cities: they all could be treated with an intelligent monitoring system that would help in the daily management of resources.

There is a lot of information there that could be used to build a more efficient city, and using new technologies to collect this data represents a massive potential to build more efficient cities. In the LS project we are building a massively scalable platform that allows to collect hundreds of streams of raw data from various agencies in real-time and process all that to infer higher-level information that can be dispatched to various agencies to optimize joint efforts. I do believe a company that is very active in this area is IBM, via their Smarter Planet initiative. I am not affiliated to IBM and the ideas I express here are only mine. But they are a great case study because they are a global company that embrace the internet of things not just as a gadget or research topic, but as a concrete product that can solve major challenges our civilization faces. Recently they partnered with the city of Rio de Janeiro to build a new operations center that operates independently of any agency while receiving data from several of them, running it through a battery of algorithms to monitor, predict, and visualize vital information in real-time to decide how best to respond, and answer optimally things such as: "Which streets will require the most troops? Which hills are most prone to mud slides? Are there shelters that have vacancies? Which hospitals have beds available? What is the best way to exit from a soccer match at the Maracana? How should officials direct traffic coming from the Copacabana Beach? Where are police cars, emergency, ambulances? Where they should go?"

Water management

Drinking water is an increasingly valuable resource throughout the world, and as cities get bigger, efficiently distributing water is becoming a major issue. Current infrastructure is aging, pipe failures are fairly common (leaks, bursts) and various reports show that on average 30% of drinkable water is lost during transmission, the system operation and management itself is often inefficient, water can be contaminated biologically or chemically. These problems have tremendous effects! First, financially it's a lot of investments and profits that go down the drain... literally. Besides, an exploding pipe in downtown costs lots of money to repair and degrades public image. Finally - and most importantly - it is a major public health concern, as in case of water shortage the energy costs to keep up with water demand will explode. Today, we know what comes in and what comes out, but the spatial and temporal resolution of data collected within the system is very low: we just know there's a leak but we don't know where and when exactly it happened.

http://aqueduct.nus.edu.sg/waterwise
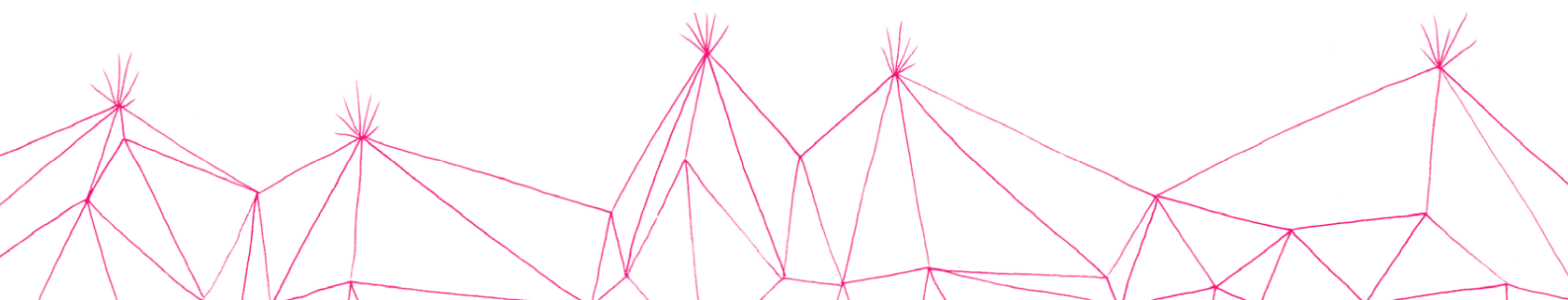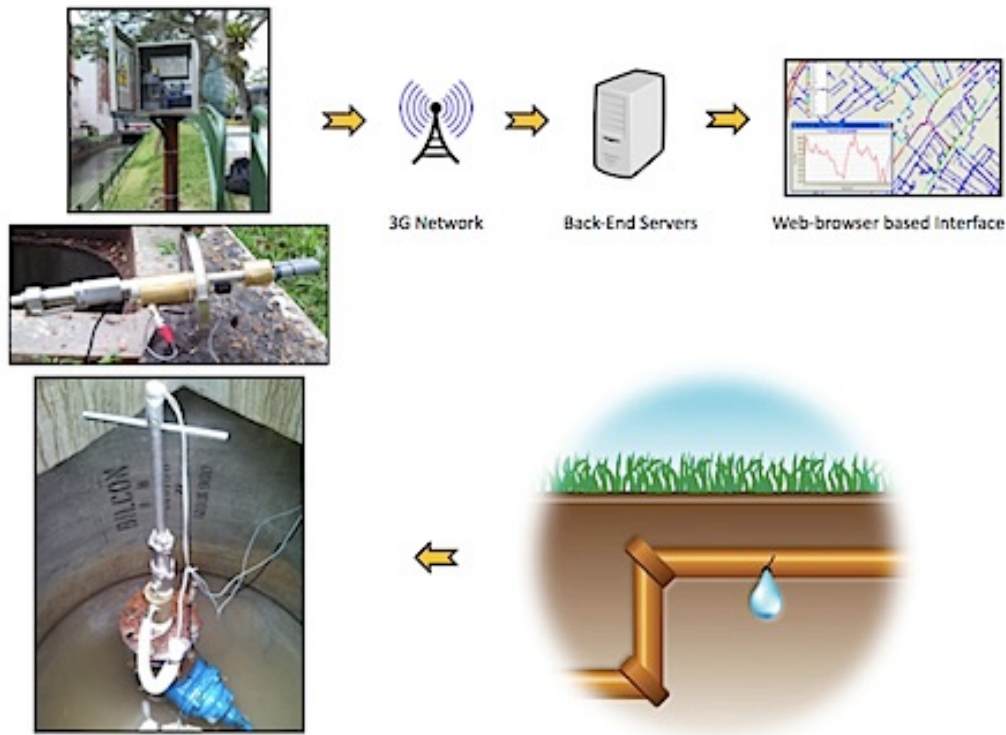
New water management strategies and technologies are a major challenge we need to address, and the sooner the better. This is another project my colleagues from CENSAM are working on in Singapore. Called Waterwise, this project aims to monitor continuously the drinking water distribution systems using sensors distributed throughout the water distribution system. Connected via a 3G connection, they are able to monitor in almost real-time the various conditions inside the pipe system such as pressure temperature, and analyze the chemical and biological composition to detect abnormalities. Firstly, the ability to quickly detect, localize leak/burst, and react quickly can reduce the amount of water lost through leakage, reduce customer disruption and minimize the extent of pipe repairs. Secondly, on-line hydraulic modeling and calibration of a water system gives an accurate, up-to-date picture of the hydraulic state of a system (flow and pressure) and the estimated consumption/demand patterns within a water distribution system.

Real-time logistics

Some people might still wonder "Ok great, but can I actually make money with this WoT thingie"? The answer is yes, and tons of it. A use case we have worked on while at SAP was how to build much more flexible BI application that tap the power and flexibility of the Web.
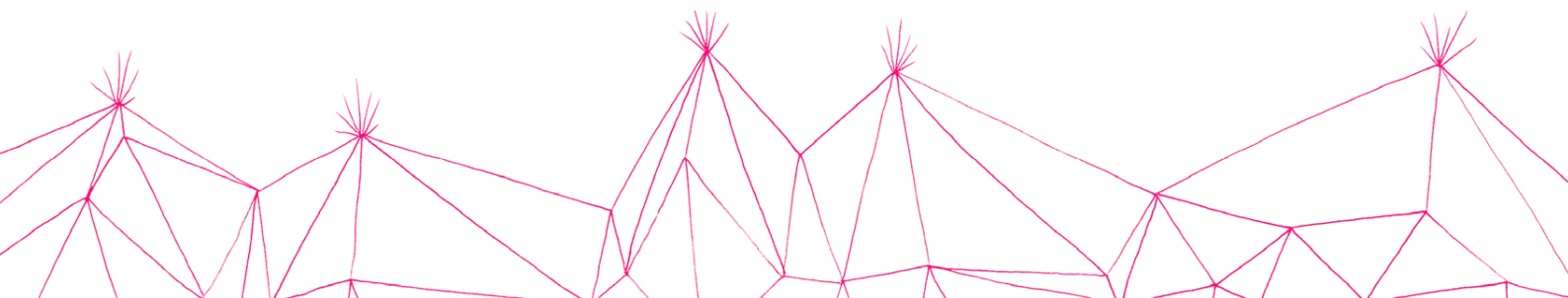
There are many important bits of information in an RFID-based supply chain, the 5W (what, when, where, who, which), and we need to integrate them efficiently and in real-time in other operations.

- The "what": what tagged products (EPCs) were read.
- The "when": at what time were the products read.
- The "where": where were the products read, in terms of Business Location (e.g. "Floor B").
- The "who": what readers (Read Point) recorded this trace.
- The "which": what was the business context (Business Step) recording the trace (e.g. "Shipping").

The EPCIS network (stands for Electronic Product Code Information System) is a set of tools and standards for tracking and sharing RFID-tagged products. It is there and used by many companies, and built by many global vendors such as SAP, oracle, IBM sell, or even the open source implementation, called Fosstrack. However, much of this data remains in closed networks and is hard to integrate. Obviously the existing products are pretty expensive and not for the average consumer.
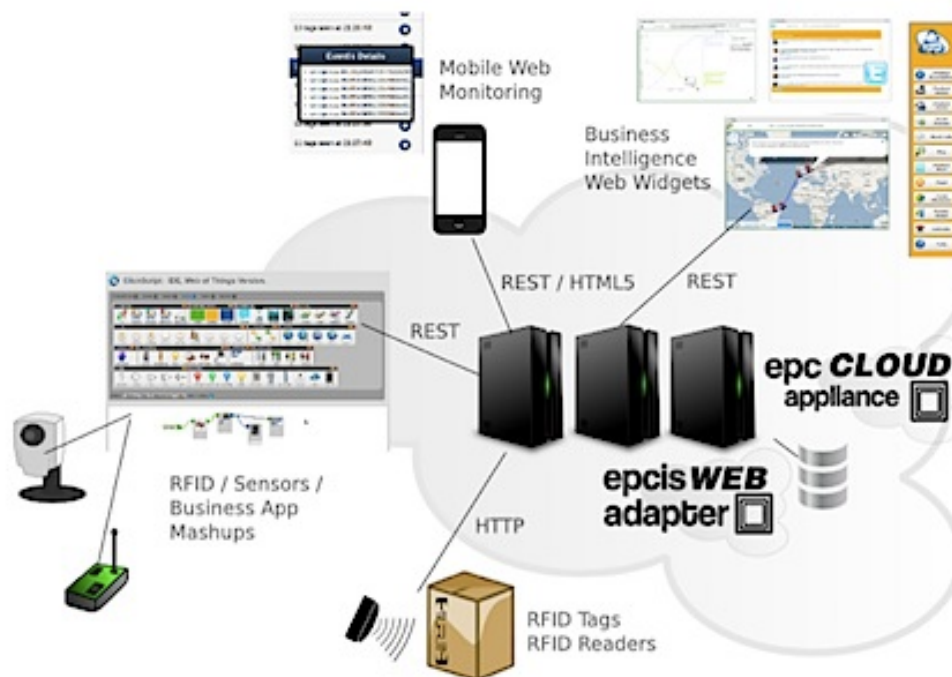
Dominique has been exploring how to make it easier to use all this data, to integrate it into various applications, and especially how to build more flexible, scalable, global application for better logistics. We create an EPC appliance in the cloud (Amazon EC2) and we we build Web APIs for accessing every standard in the cloud.
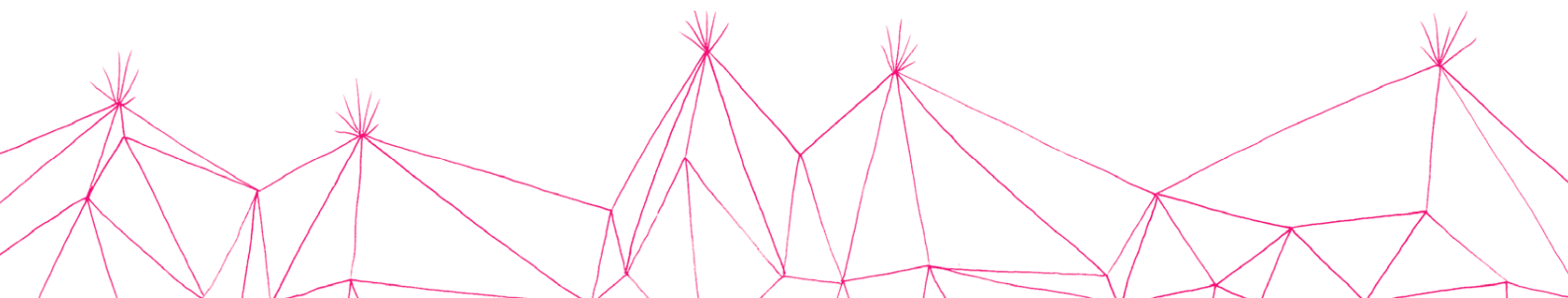
With this, one can create:

1. a scalable, distributed DB for the RFID events (readings),
2. Business intelligence widgets using the EPCIS REST API,
3. Mobile Web Apps that can be used to monitor readers in-situ (HTML5 push for the actual implementation),
4. RFID / Sensors / Business apps (e.g., ERP) Web mashups (even people can do that, e.g., what to do when an object gets stolen?)

Fears

Over the years, we have received much criticism about building such an Orwellian society where everything is interconnected, tagged, tracked, monitored. I can't deny these fears are justified, but just like with any other technology it's not good or bad. It's essential to put in the place not only the technical, but also legal barriers to minimize misuse of all this data. So let me ask you something: Are your e-banking transactions publicly available over the web? no. Are your emails publicly available online? No, unless you hit the notorious "reply all" button. Can I access your company's intranet? Of course not.

We have been building sufficiently secure websites for almost two decades, and in the Web of Things, these well-known mechanisms would come for free. I'm not insinuating these are perfect, but that they are good enough for most use cases. So why not just leverage them for physical devices?

Conclusions

1. WoT is here and it matters, but it is not the best solution for everything.
2. Think about niwea when you think mobile
3. Stop talking, start prototyping

I hope that I've been able to explain what is the Web of Things, why is it different from the Internet of Things, and in what this difference matters. Hopefully, as more people understand the value and potential of sharing data for not only themselves, but for all of us, we'll see more such projects emerge. This in turn will accelerate innovation and allow us to build more efficient and sustainable cities, without sacrificing the quality of life.

We would like to thank once more here all the people who helped and supported us (financially and psychologically) throughout our research. Of course our professor Friedemann Mattern and our colleagues and students at the distributed systems group at ETH Zurich. Then SAP research and the EU projects socrades and sensei that have funded half of our phd. Then you all, the community that inspired us and reminded us that what we do actually matters, you're all great!

www.webofthings.com